CSC 249/449 Machine Vision: Homework 4
Term: Spring 2019
Instructor: Prof. Chenliang Xu
TA: Yapeng Tian, Jing Shi, Zhiheng Li, Yutong (Kelly) He, Chaoying Xue
Due Date: 3/30/2019 Saturday 11:59pm
Constraints: This assignment is to be carried out independently.

Problem 1 (100 pts): Complete an image captioning network

Implement the forward pass of a decoder network for image captioning and a one-layer LSTM unit. The project is in https://github.com/YapengTian/csc249_hw4. Complete the code marked by **TODO** in *network.py*.

You can refer to this pytorch document (https://pytorch.org/docs/stable/nn.html/) to know more details about convolutional layer, fully-connected layer, RNN layer, activation functions, and loss functions.

Do **NOT** modify function interfaces. If you want to add parameters to a function, please provide default values so that the original behavior of the function is unchanged.

Before running the code please first follow *ReadME.md* to install the dependencies.

1. (40 pts) LSTM-based Language Decoder

   In this assignment, you will build a LSTM-based language decoder network based on the defined layers in the *network.py*. The decoder mainly consists of three layers: a word embedding layer, a one-layer LSTM, and a linear (fully-connected) layer. We have already defined them in the *init* function of the Decoder class. You will implement the *forward* function using the defined layers. The network structure can be found in *image captioning tutorial* given in the class and the computation steps are also described in the comments of the code.

   - Implement the *forward* function of the Decoder class.
   - Run *python train.py* to train an image captioning model based on the implemented Decoder.
   - Run *python predict.py* to produce a caption for a bird image and test other images by running *python predict.py —image image_path*.
   - Compute BLEU, METEOR, ROUGE_L, and CIDEr scores for 100 testing images by running *python eval.py*. After one epoch of training, the model should be able to achieve around 0.65 on CIDEr.
   - Record the experiments in your report.

2. (40 pts) Implement Your LSTM

   In this assignment, you will implement your LSTM unit based on the following formulations:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \tag{1}$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \tag{2}$$
$$\widetilde{C_t} = tanh(W_C \cdot [h_{t-1}, x_t] + b_C), \tag{3}$$
$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t}, \tag{4}$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \tag{5}$$
$$h_t = o_t * tanh(C_t). \tag{6}$$

   - Implement the *LSTMCell* class in the *network.py* with the above equations. Given inputs: x_t, (h_{t-1}, C_{t-1}), the *LSTMCell* will return (h_t, C_t).

- Test your implementation by replacing the *nn.LSTM* from Pytorch with the *LSTM*, which is based on your implemented *LSTMCell*, and then re-train your image captioning model. You can compare the CIDEr score of the model with your previous one trained with *nn.LSTM* by running *python eval.py*.
- Record the experiments in your report.

3. (10 pts) Compete against Microsoft's AI

   Explore and compare to an open caption bot. Of course, you do not need to beat the Microsoft's caption bot.

   - Select a few images you like and use both your model and the Microsoft's caption AI: https://www.captionbot.ai/ to generate captions.
   - Which one is better? Explain why.
   - Record the experiments in your report. Your report should contain 3 selected images and generated sentences.

4. (10 pts) New ideas

   Come up with new ideas on how to improve the image captioning model. This is an open problem. You can obtain answers either from research papers or from your experimental findings. It can be some tricks for training or more advanced network structures.

Problem 2 (20 pts): Evaluation Metrics for Image Captioning (**Graduate ONLY**)

There are four commonly used evaluation metrics: BLEU, METEOR, ROUGE_L, and CIDEr. In this assignment, you will select two metrics to write their definitions and discuss the limitations.

- Papineni *et al.* BLEU: a Method for Automatic Evaluation of Machine Translation, ACL, 2002.
- Banerjee *et al.* METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments , ACL workshops, 2005.
- Lin *et al.* ROUGE: A Package for Automatic Evaluation of Summaries, WAS, 2004.
- Vedantam *et al.* CIDEr: Consensus-based Image Description Evaluation, CVPR, 2015.

Submission Process: Please follow the submission instruction.

The submitted .zip file should be named with your netID: your_netID.zip. It should only contain the following files:

- *network.py*
- *captions_res.json* generated by running *python eval.py* with your implemented LSTM-based image captioning model
- Your report named with your netID: your_netID.pdf.